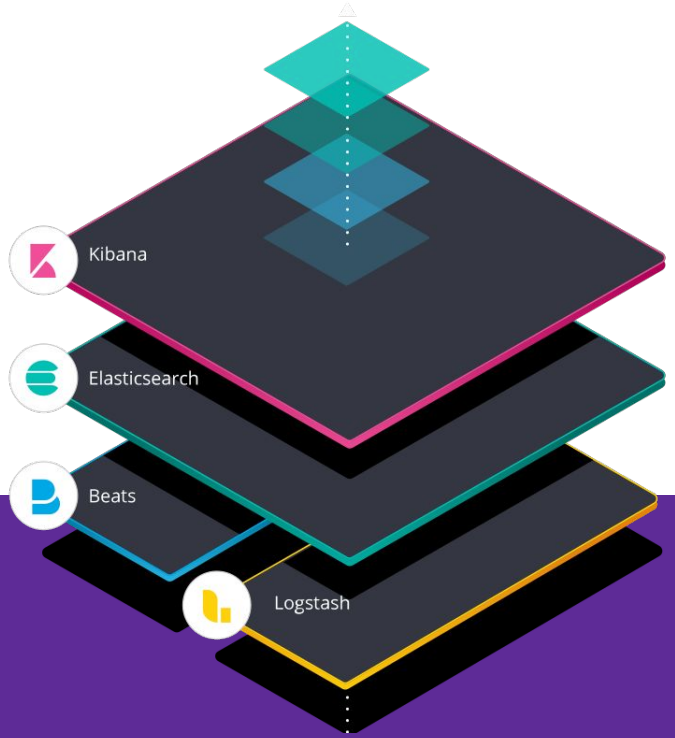
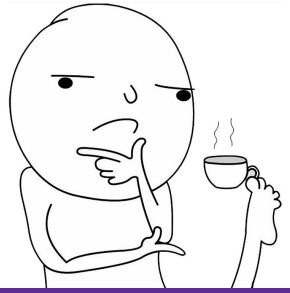


ELK

Agenda

- 🔨 The story of logging
- 🔨 How does ELK works!
- 🔨 Key features
- 🔨 Demo
- ⛳ Hand-on use case

The story of logging



Application logging

Logging sensitive information

Example Language: **Java** *(bad code)*

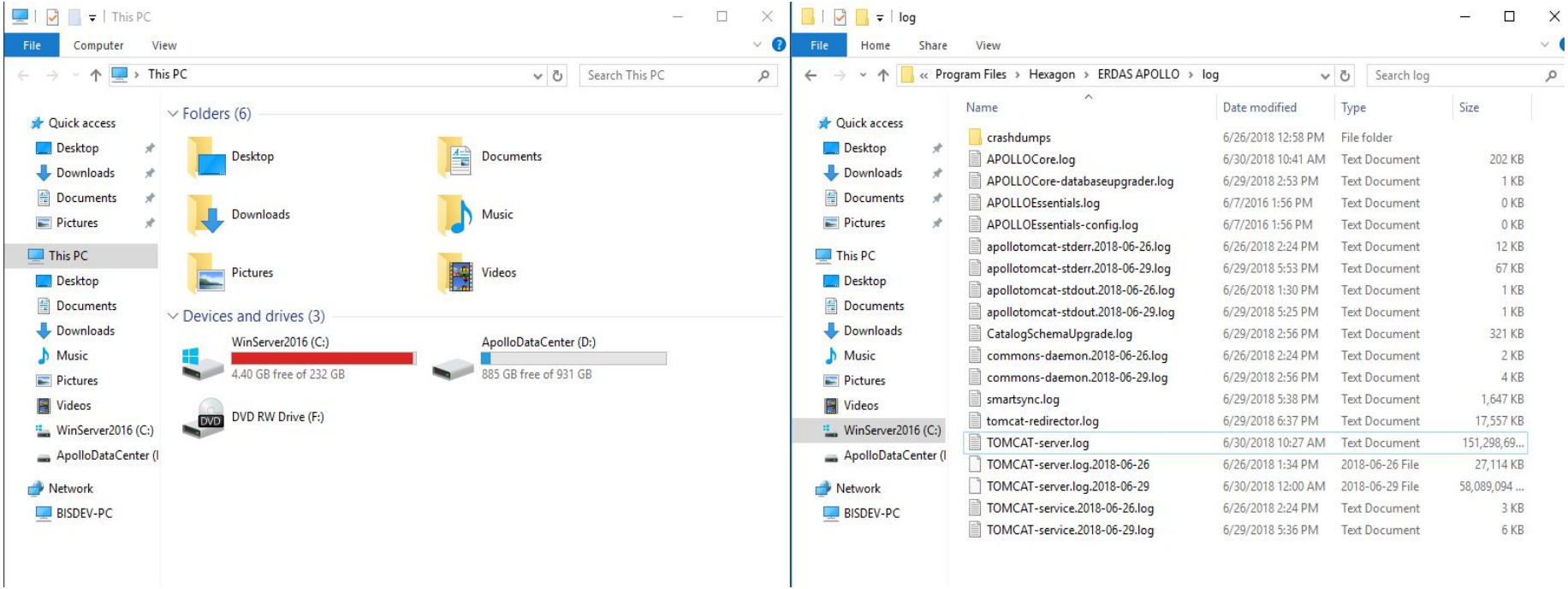
```
logger.info("Username: " + usernme + ", CCN: " + ccn);
```

Example Language: **Java** *(bad code)*

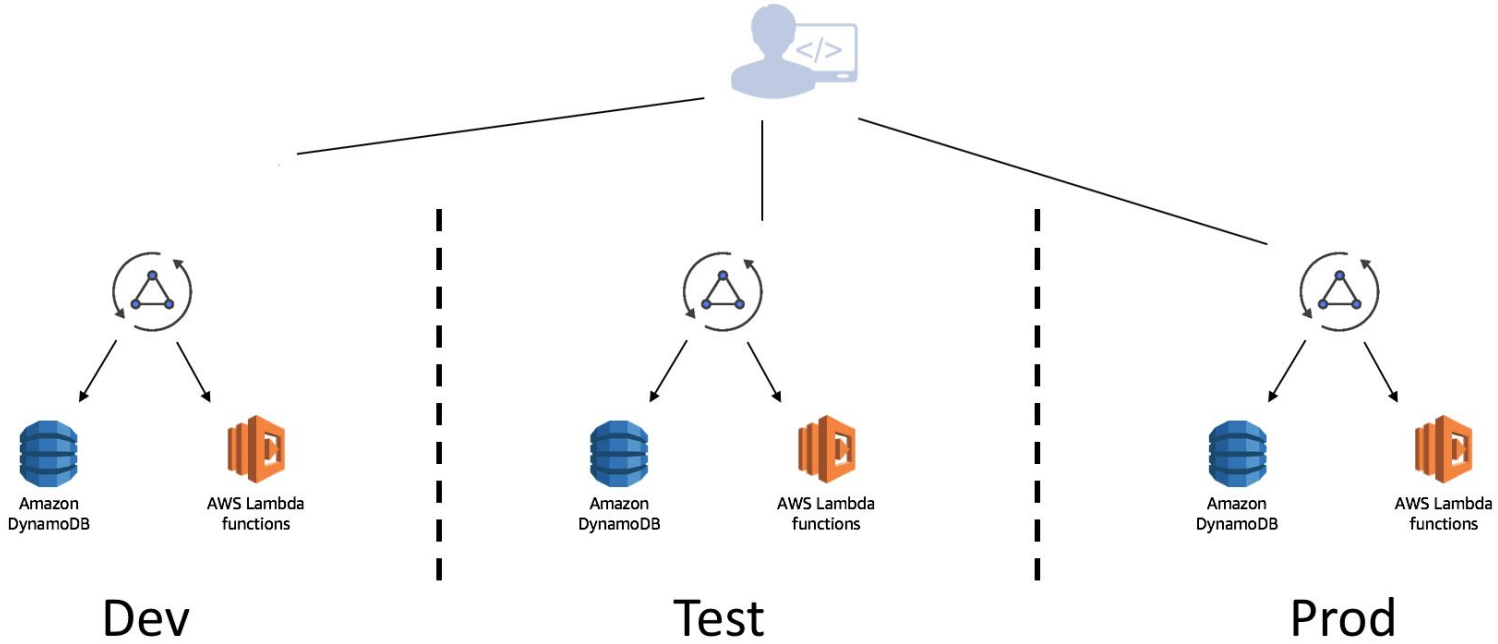
```
locationClient = new LocationClient(this, this, this);
locationClient.connect();
currentUser.setLocation(locationClient.getLastLocation());
...

catch (Exception e) {
  AlertDialog.Builder builder = new AlertDialog.Builder(this);
  builder.setMessage("Sorry, this application has experienced an error.");
  AlertDialog alert = builder.create();
  alert.show();
  Log.e("ExampleActivity", "Caught exception: " + e + " While on User:" + User.toString());
}
```

Using a single log file



Monitoring across multiple environments



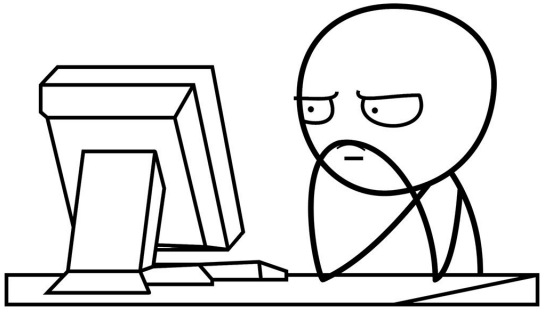
ELK Stack

ElasticSearch: storing logs

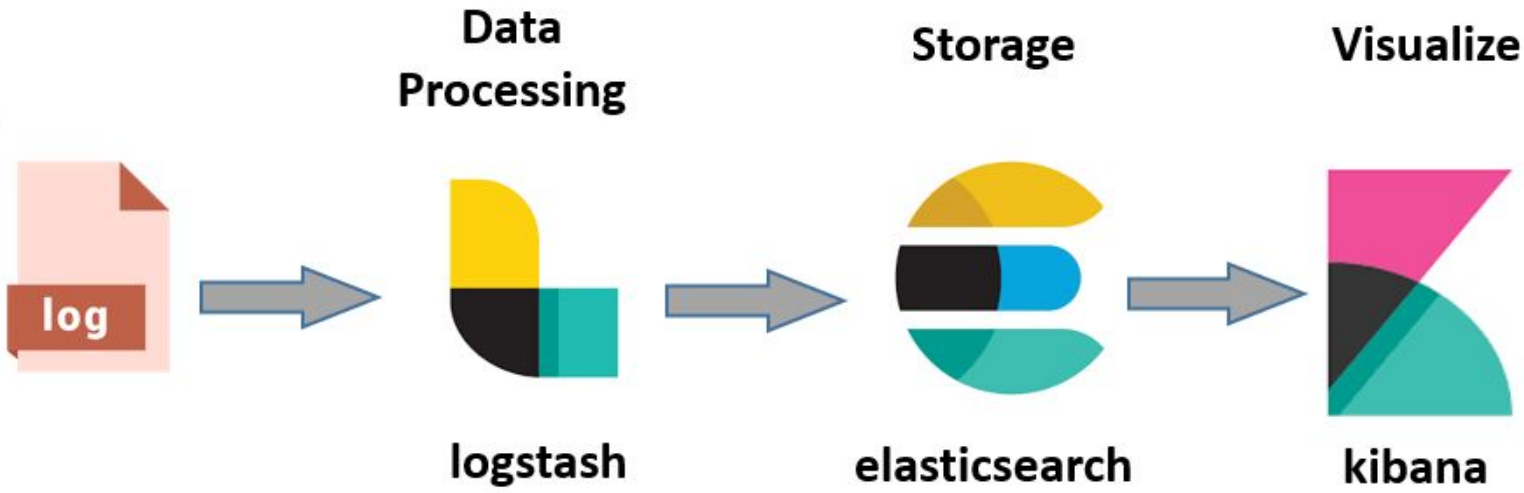
LogStash: processing logs

Kibana: visualization tool

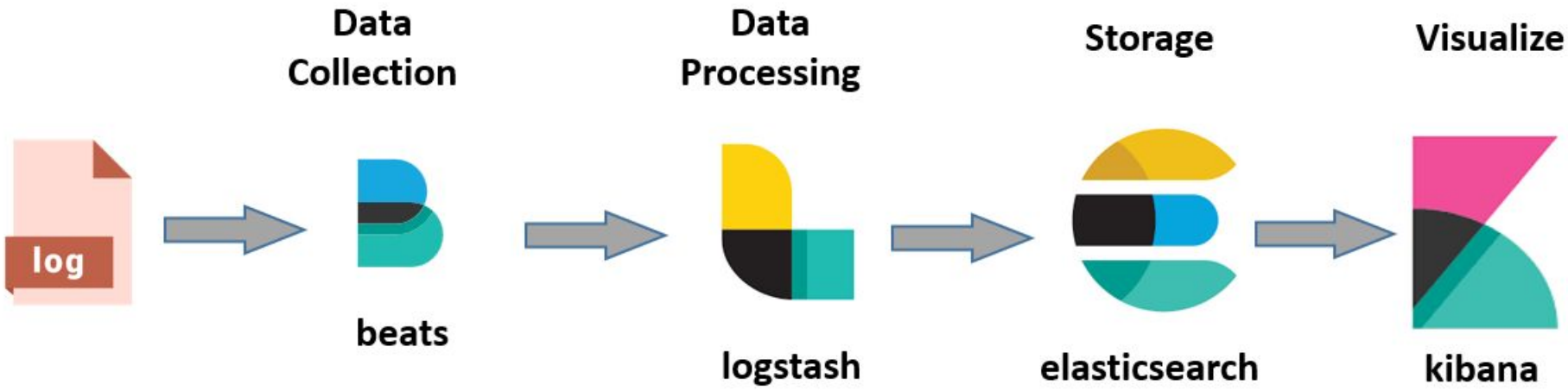
How does ELK work!



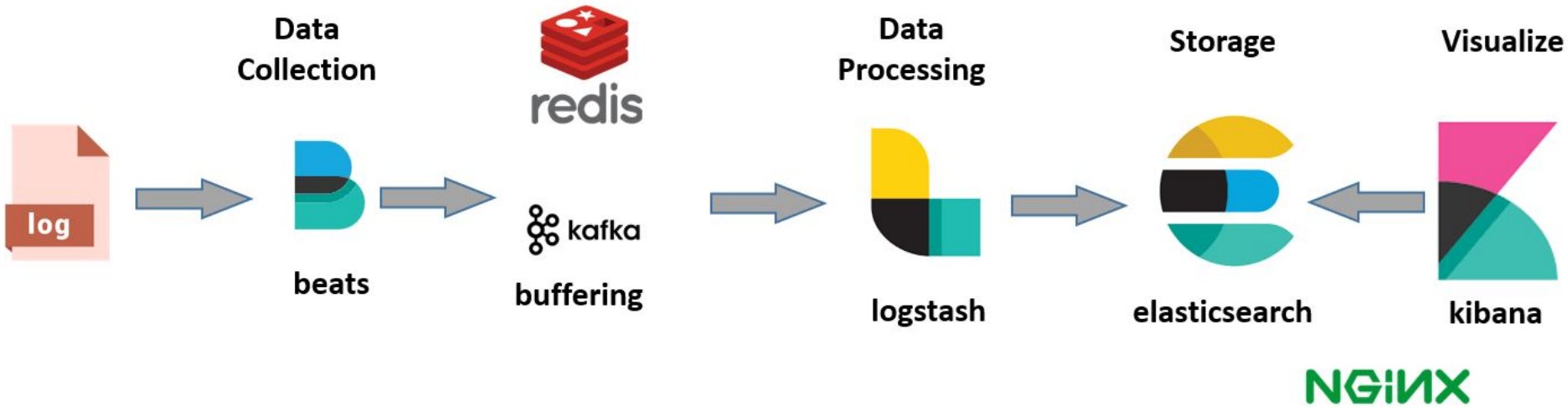
Classic architecture



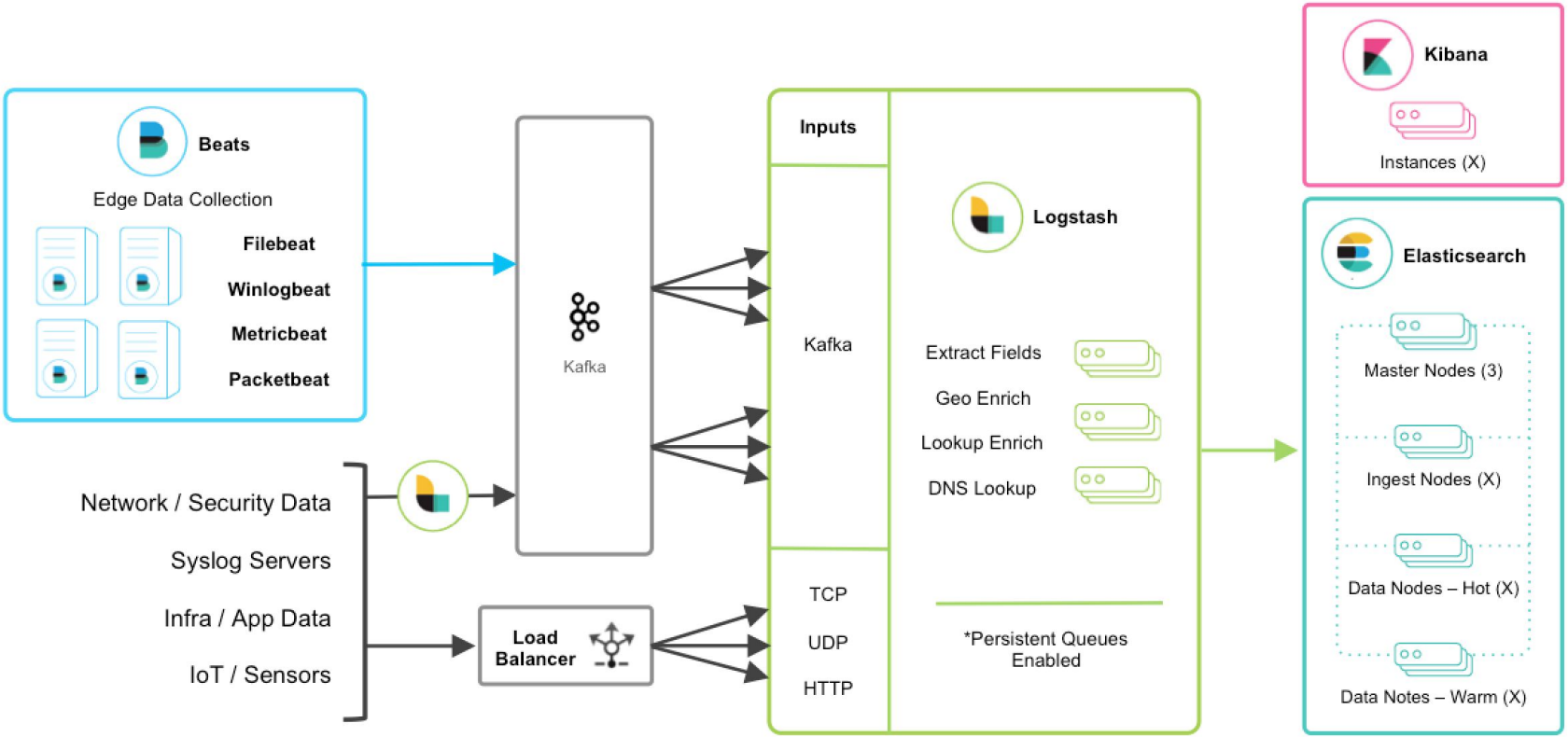
Elastic stack architecture



Massive amount of data architecture



Data flow


Beats
Edge Data Collection
Filebeat
Winlogbeat
Metricbeat
Packetbeat
Network / Security Data
Syslog Servers
Infra / App Data
IoT / Sensors
Kafka
Load Balancer
Inputs
Kafka
TCP
UDP
HTTP
Logstash
Extract Fields
Geo Enrich
Lookup Enrich
DNS Lookup
*Persistent Queues Enabled
Kibana
Instances (X)
Elasticsearch
Master Nodes (3)
Ingest Nodes (X)
Data Nodes – Hot (X)
Data Notes – Warm (X)

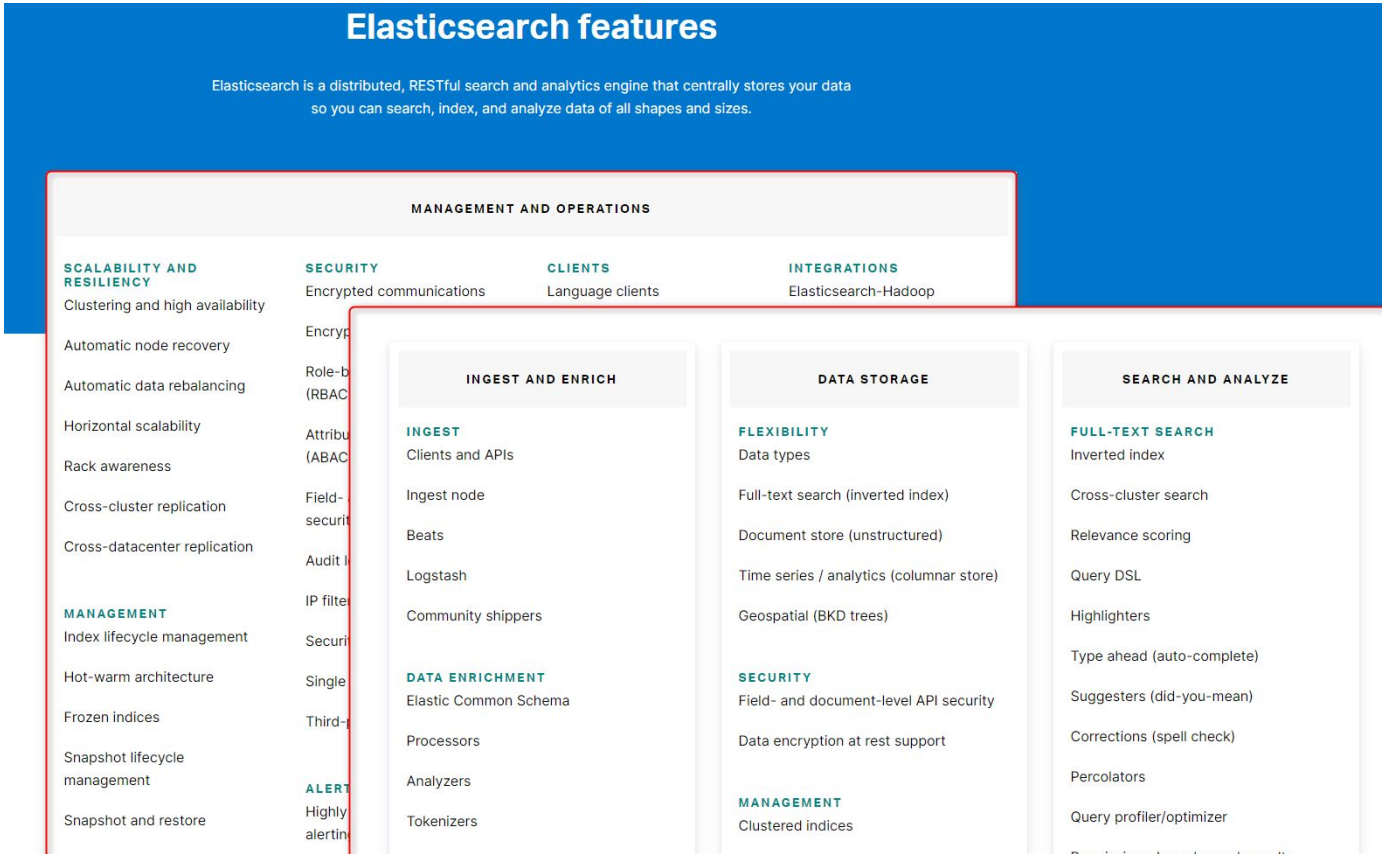
Case studies

Advantages and Disadvantages

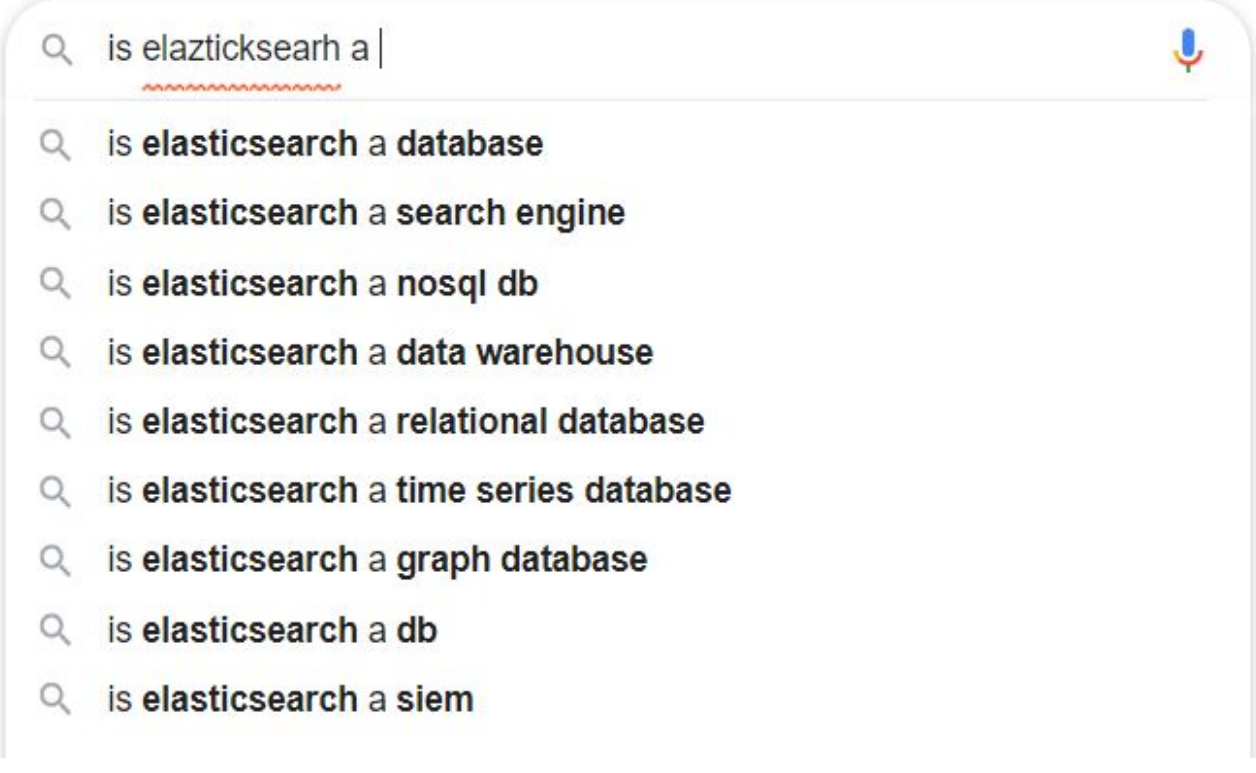
Advantages	Disadvantages
(1) Works best when logs from various apps (2) Rapid installation (3) Scales up vertically and horizontally (4) Offers a host of language clients	(1) Different components in the stack can become difficult

Elasticsearch

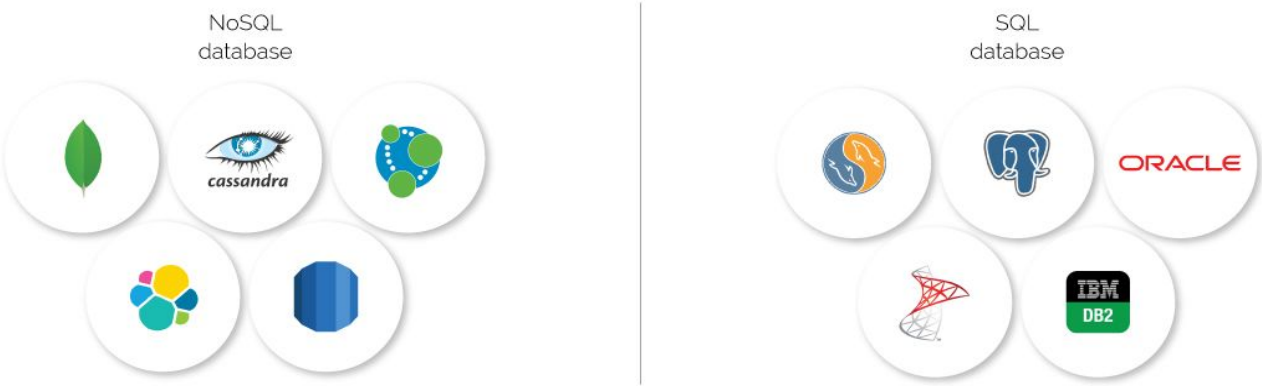
Elasticsearch key features



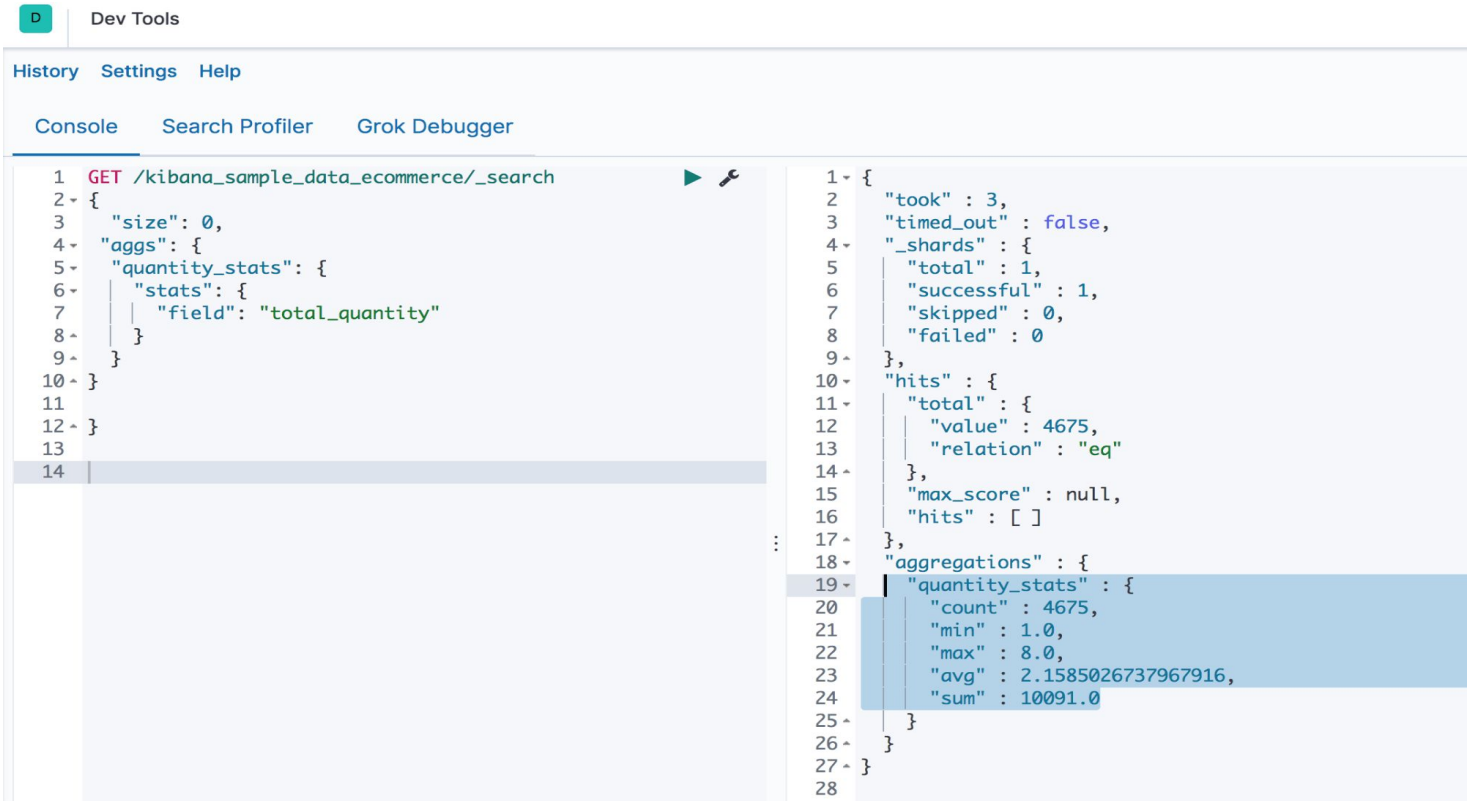
Full text - powerful search engine



A replacement of NoSQL DBMS



Query & Analyze structured data



Analyze log and system metric

Fundamental concepts (1/3)

Index: Elasticsearch Indices are logical partitions of documents, or a set of documents. (RDBMS: table)

Documents: Documents are JSON objects that are stored within an Elasticsearch index and are considered the base unit of storage. (RDBMS: record)

Fundamental concepts (2/3)

Types: Elasticsearch types are used within documents to subdivide similar types of data wherein each type represents a unique class of documents.

(RDBMS: SQL data type)

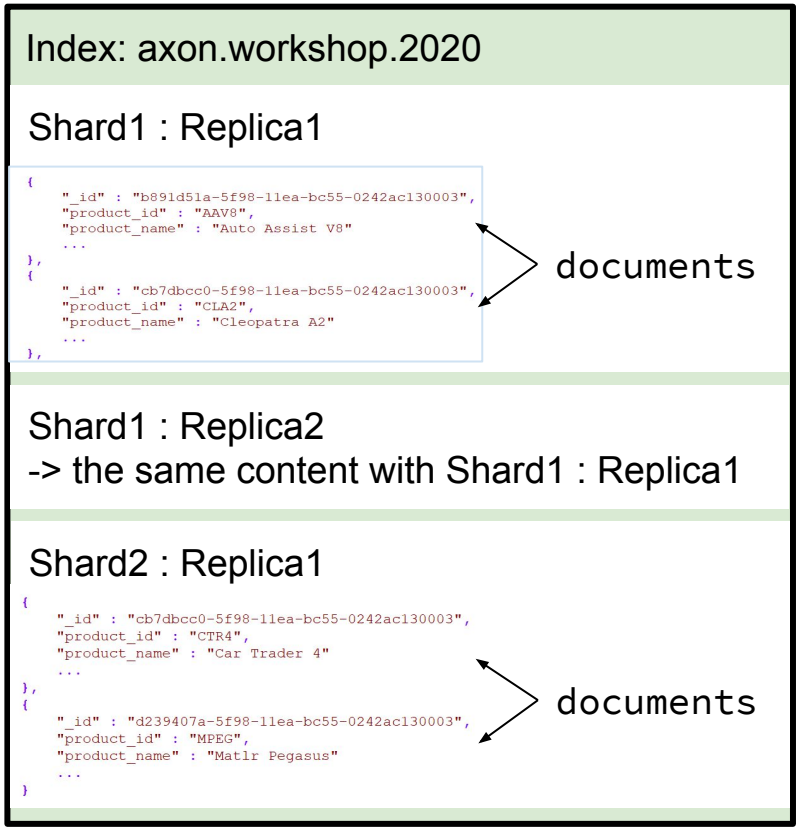
Mapping: Mapping defines the type of different fields that reside within an index.

Fundamental concepts (3/3)

Shards: Index size is a common cause of Elasticsearch crashes
-> Split up indices horizontally into pieces called shards.

Replicas: Elasticsearch allows users to make copies of shards called replicas.

Fundamental concepts in pictures




```
# PUT /axon.workshop.2020
{
 "mappings": {
       "properties": {
        "product_id":    { "type": "keyword" },
        "product_name":  { "type": "text"  },
        "price":  { "type": "float"  },
        ...
       }
 }
}
```

Elasticsearch queries

Boolean Operators

- jack AND jill — return events contain both jack and jill
- ahab NOT moby — return events contain ahab but not moby
- tom OR jerry — return events contain tom or jerry, or both

Elasticsearch queries

Fields

- Name:“Ned Stark”

Elasticsearch queries

Ranges

- age:[3 TO 10] — Will return events with age between 3 and 10
- price:{100 TO 400} — Will return events with prices between 101 and 399

Elasticsearch queries

Wildcards, Regexes and Fuzzy Searching

- You can use the * character for multiple character wildcards or the ? character for single character wildcards.

Elasticsearch queries

URI Search

- curl "localhost:9200/_search?q=name:travis"
- curl "localhost:9200/_search?q=name:john~1 AND (age:[30 TO 40} OR surname:K*) AND -city"

Elasticsearch REST API

- **Elasticsearch Document API**

```
curl -X PUT "localhost:9200/twitter/_doc/1?pretty" -H
'Content-Type: application/json' -d'

{  "user" : "kimchy",

  "post_date" : "2009-11-15T14:12:12",

  "message" : "trying out Elasticsearch"  }'
```

Elasticsearch REST API

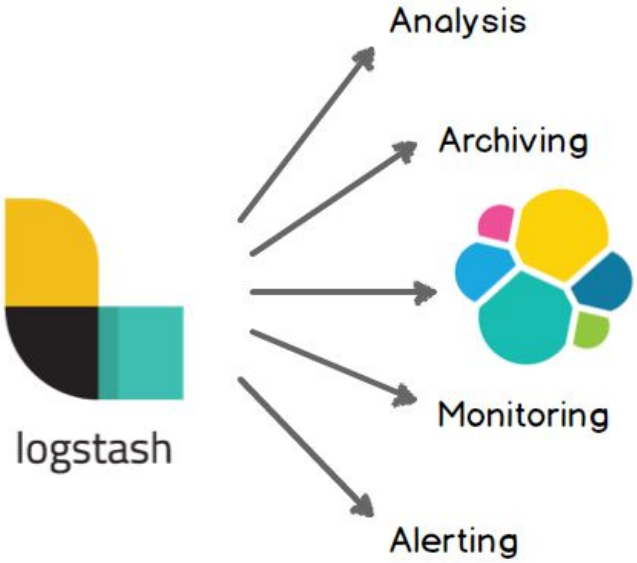
- Elasticsearch Search API
- Elasticsearch Indices API
- Elasticsearch Cluster API

Elasticsearch Plugins

Elasticsearch plugins are classified in two types - core plugins or community plugins

Logstash

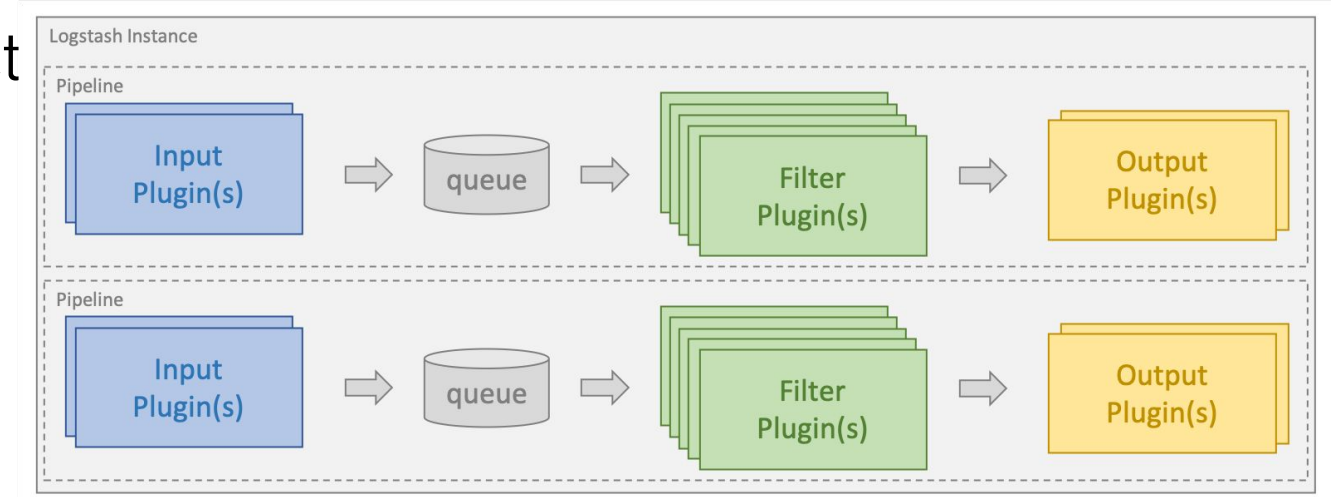
Logstash key features



- Collect from different sources.
- Filters, parsing, transforming meaning data.
- Multiple destinations.

Logstash key concepts

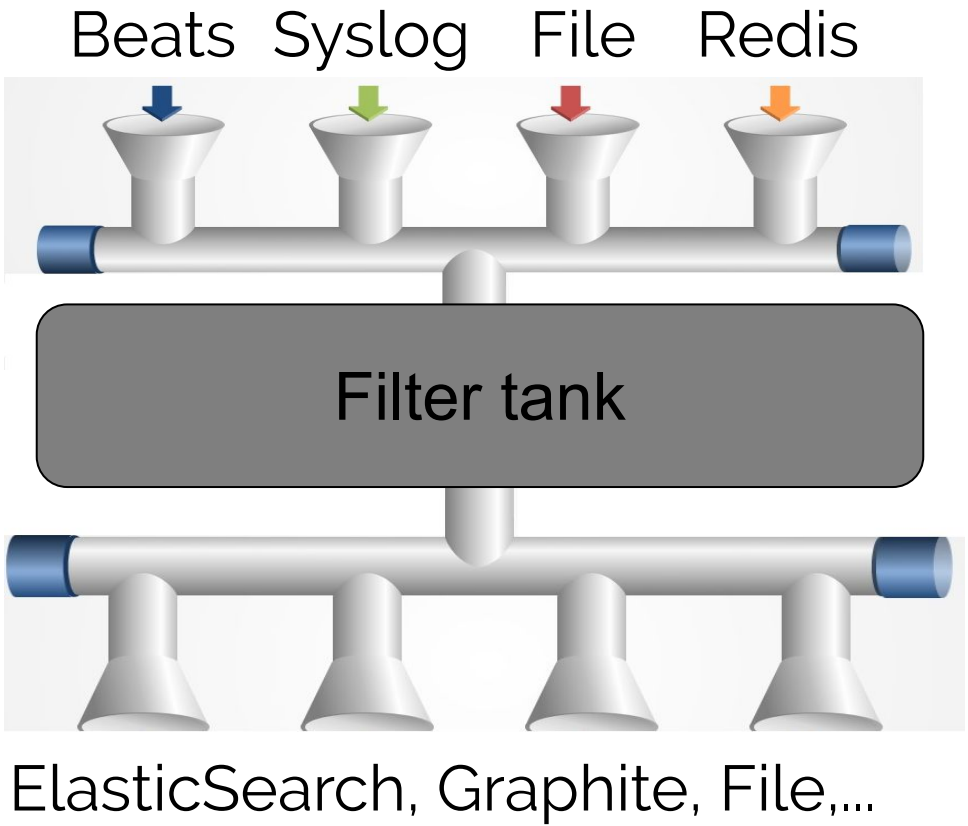
- Event object
- Pipeline
- Input
- Filter
- Output



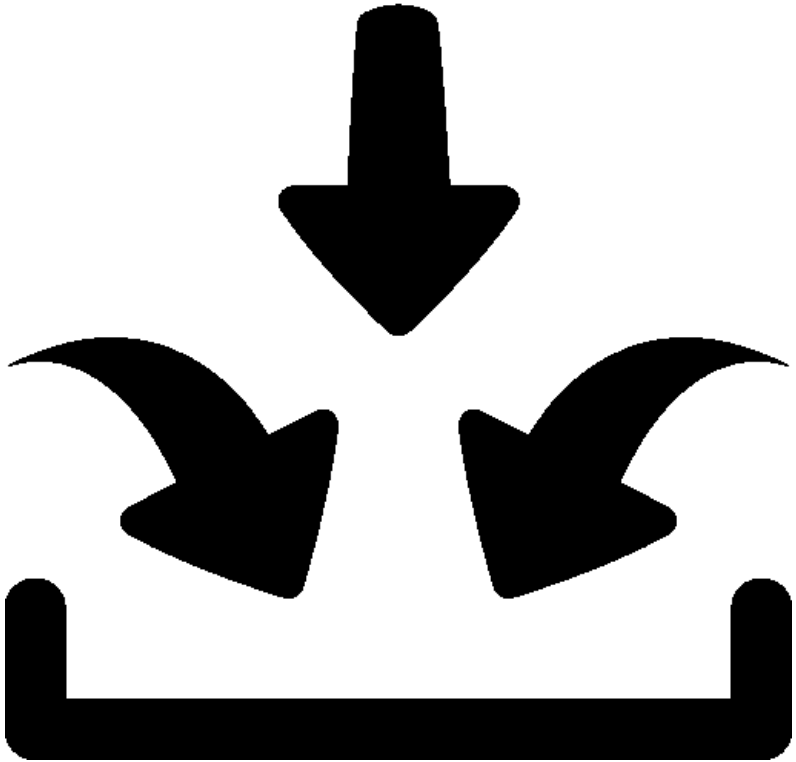
Event object

```
{
       "message" => "Hello Logstash!",
      "@version" => "1",
          "host" => "192.168.1.69",
    "@timestamp" => 2020-02-15T03:56:41.001Z,
       "headers" => {
         "request_method" => "PUT",
           "http_version" => "HTTP/1.1",
            "http_accept" => "*/*",
          "content_type" => "text/plain;charset=UTF-8",
       "accept_language" => "vi,en;q=0.9,en-US;q=0.8,de-CH;q=0
       "http_user_agent" => "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
            "connection" => "keep-alive",
         "postman_token" => "a8581dcb-abfc-0249-8706-500a1064120e",
         "cache_control" => "no-cache",
        "content_length" => "15",
       "accept_encoding" => "gzip, deflate",
          "request_path" => "/myAppId",
                "origin" => "chrome-extension://fhbjgbiflinjbdggehcddcbncdddomop",
           "http_host" => "192.168.1.4:5044"
    }
}
```

Pipeline

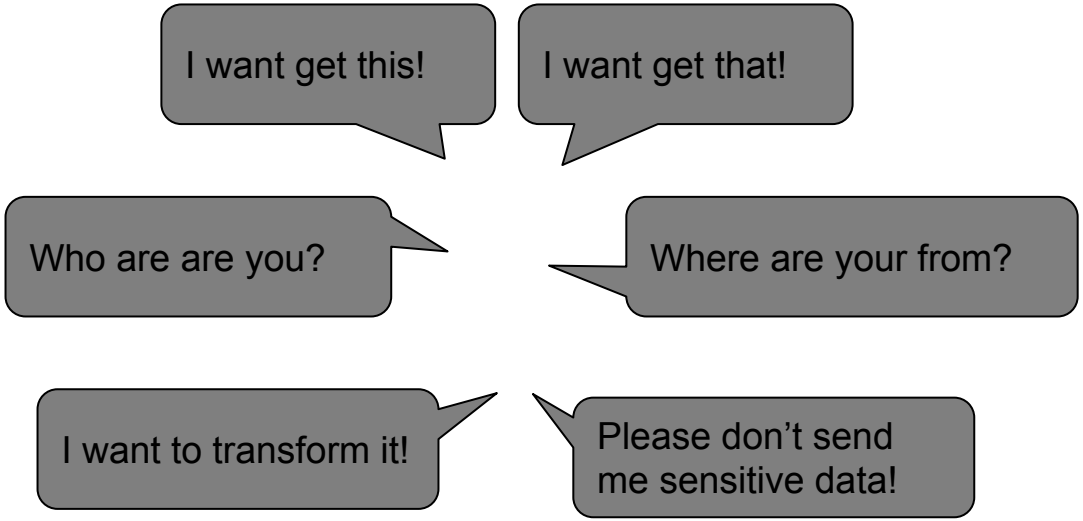

Beats
Syslog
File
Redis
Filter tank
ElasticSearch, Graphite, File,...

Input



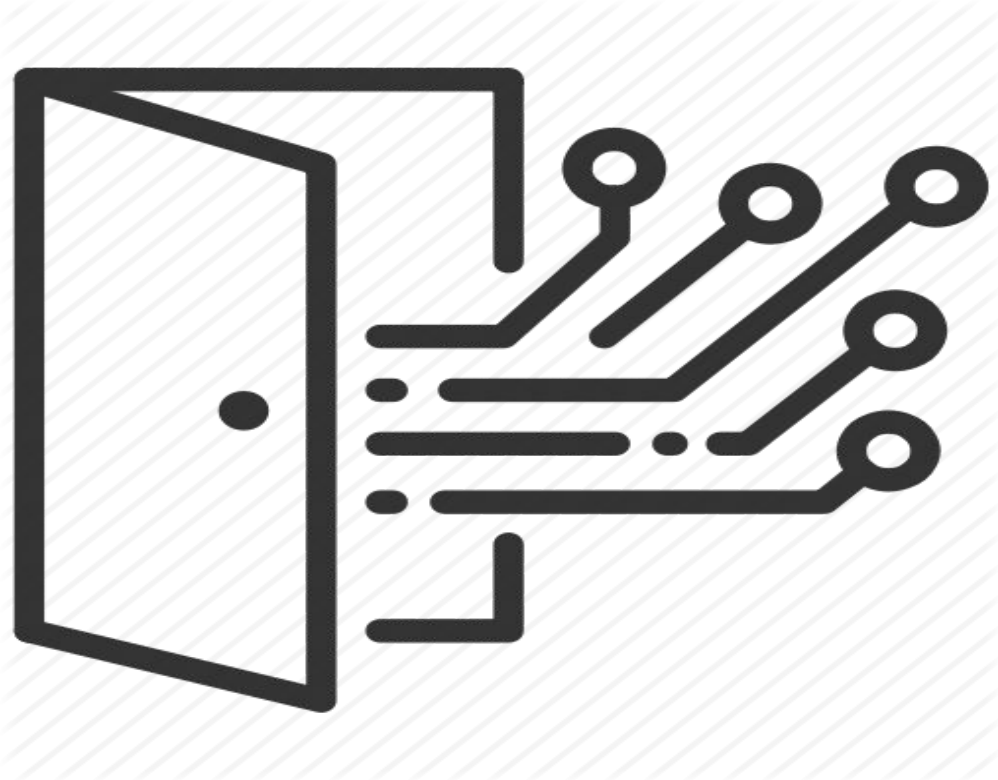
- Beats
- Syslog
- File
- Redis
- ...

Filter



- Grok
- Mutate
- Drop
- Geoip
- ...

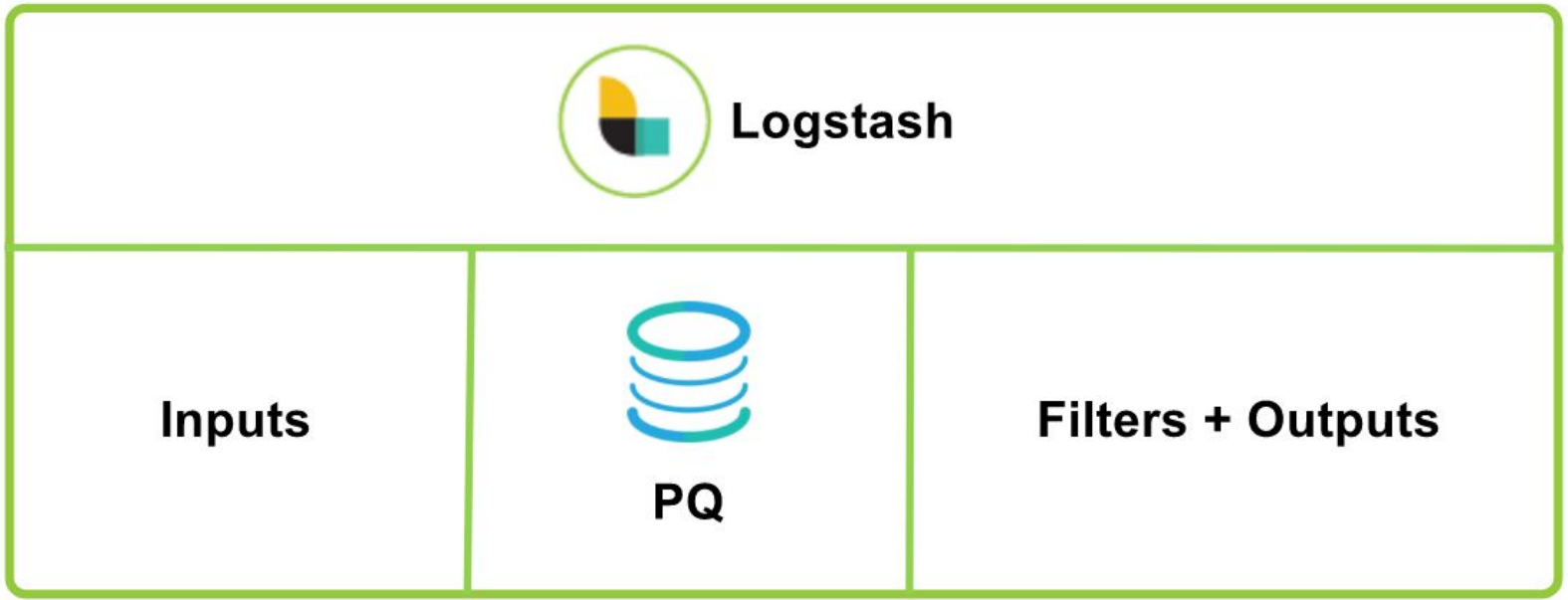
Output



- Elasticsearch
- File
- Graphite
- ...

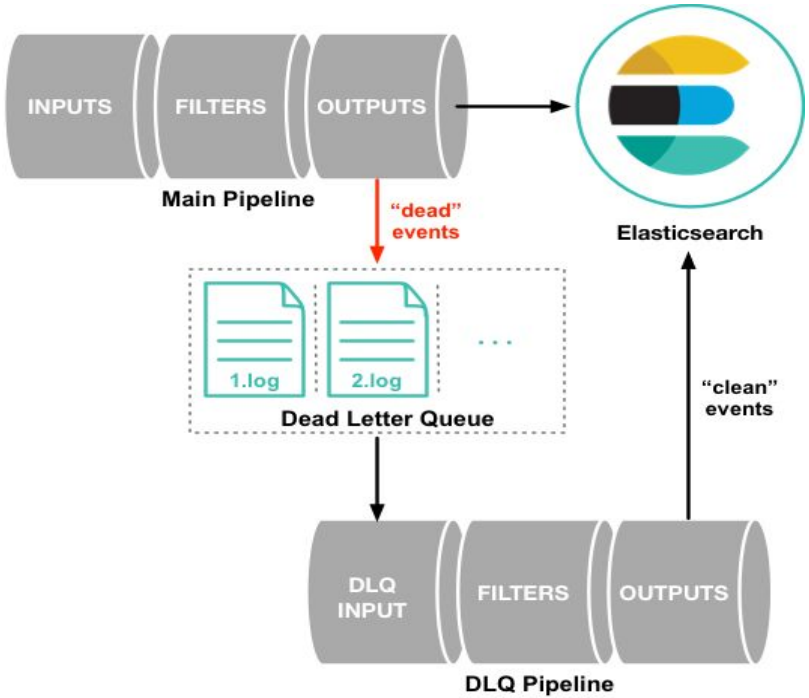
Data Resiliency

- Persistent Queues



Data Resiliency

- Dead Letter Queues (output)



Logstash pros and cons

Pros	Cons
(1) Regex pattern (2) Supports a variety flatform (3) Bunch of plugins	(1) Need good understanding (2) Filter plugins not generic

Configuration

```
input {
    beats {
        port => "5044"
    }
}
filter
    grok {
        match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    date {
        match => ["timestamp", "dd/MMM/yyyy:HH:mm:ss Z"]
    }
    geoip {
        source => "clientip"
    }
}
```

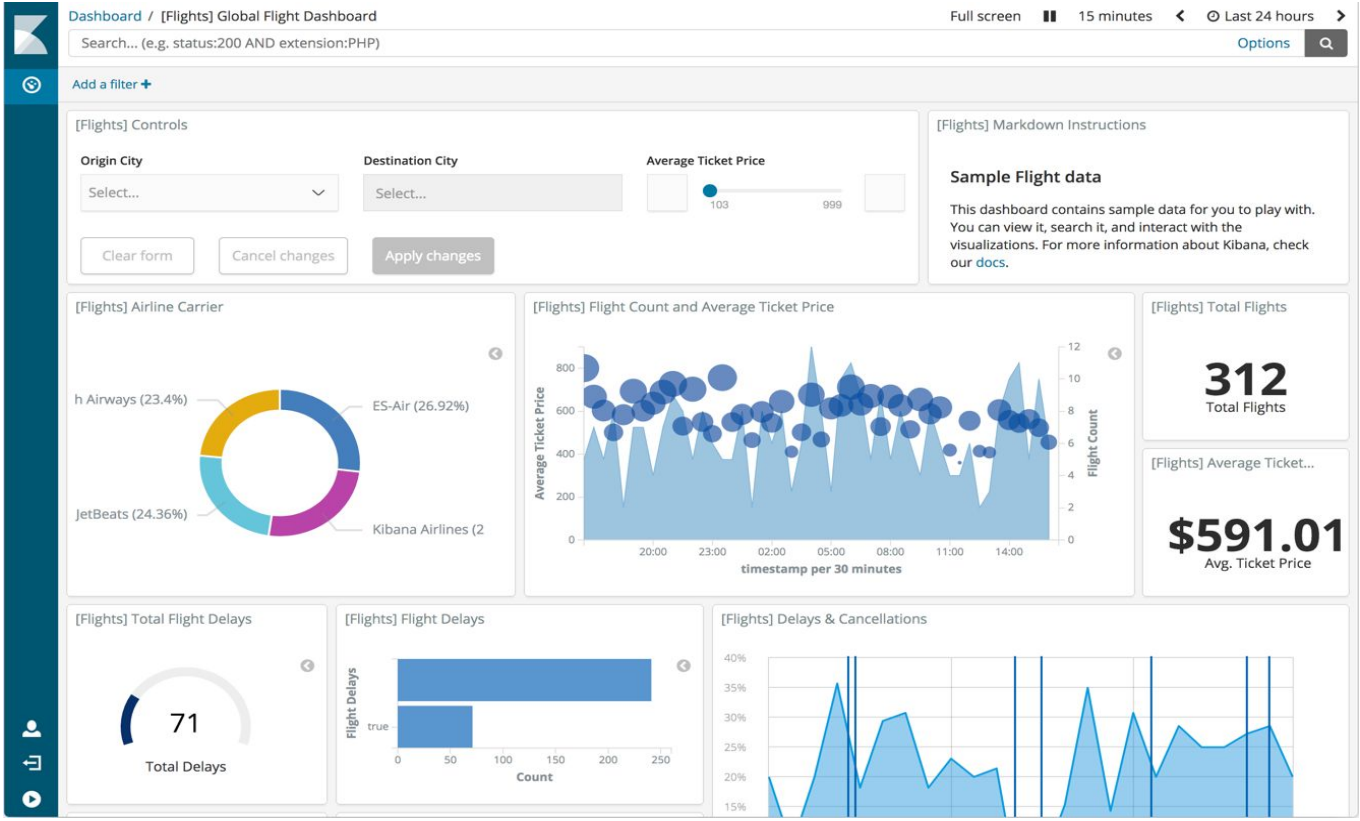
```
output {
    elasticsearch {
        hosts ["localhost : 9200"]
    }
}
```

Kibana

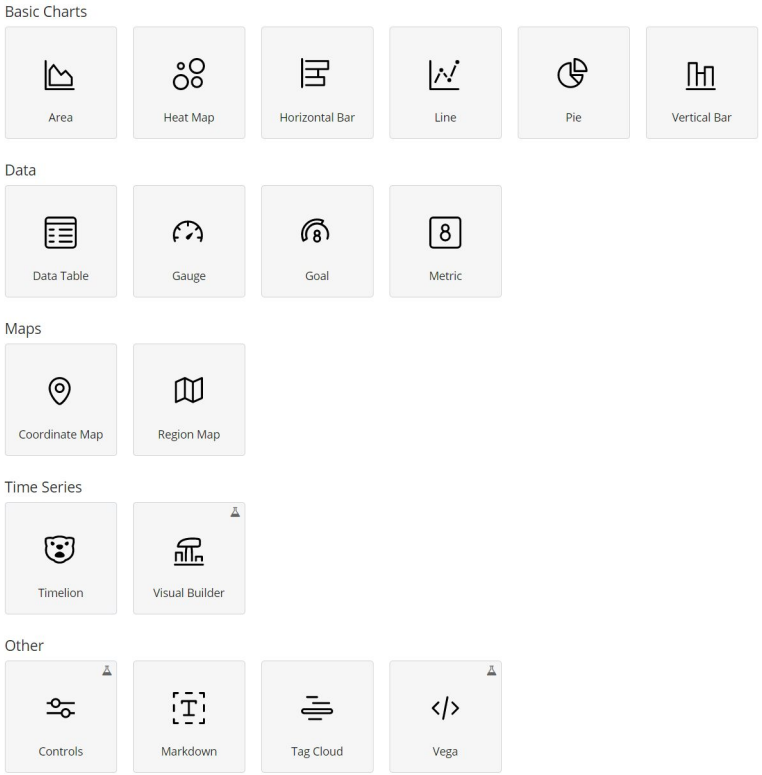
Kibana key features

- Elasticsearch documents visualization by various of methods (Chart, Metric,...)
- Monitoring (Alert, Log,...)

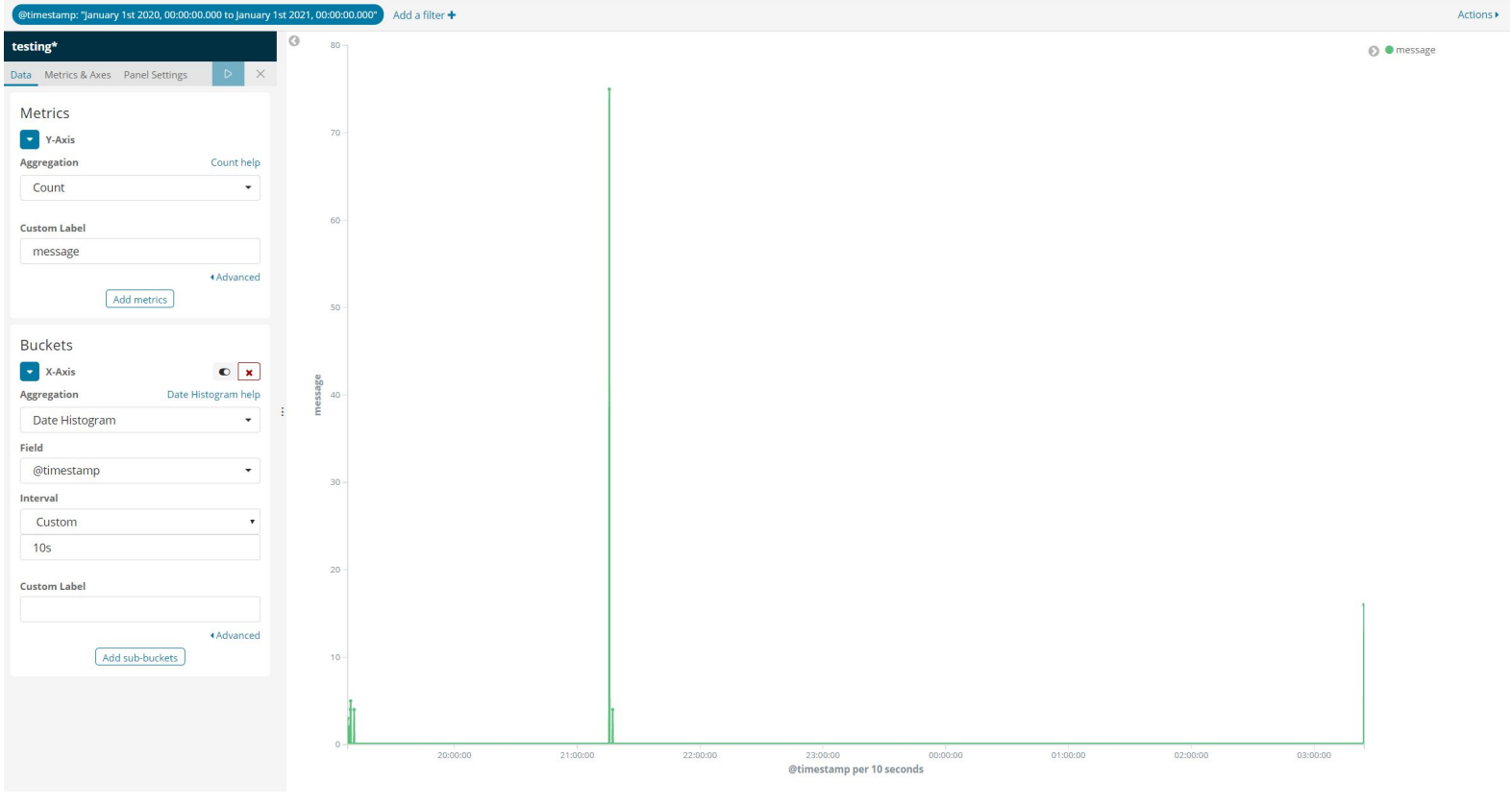
Data visualization



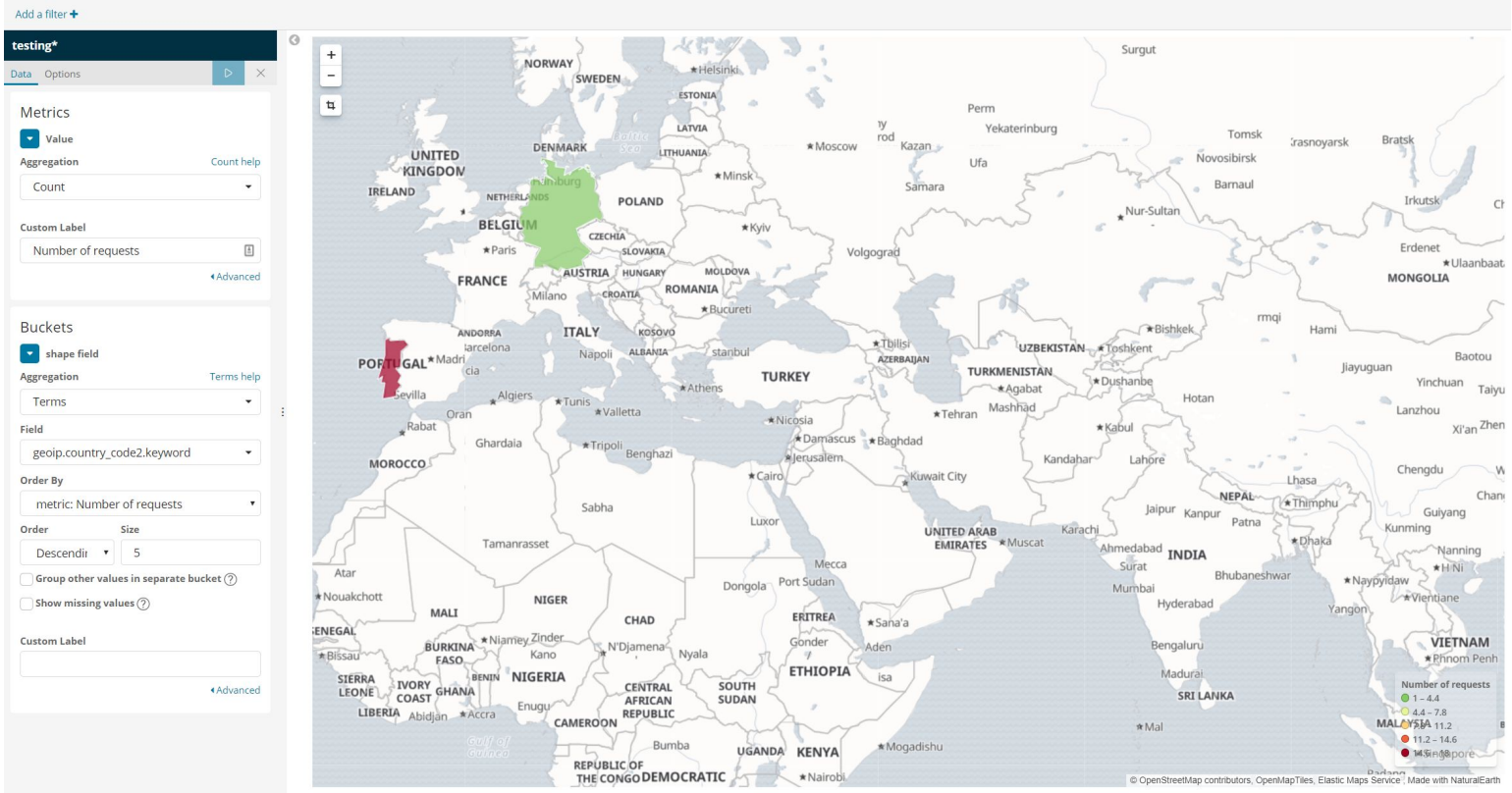
Data visualization



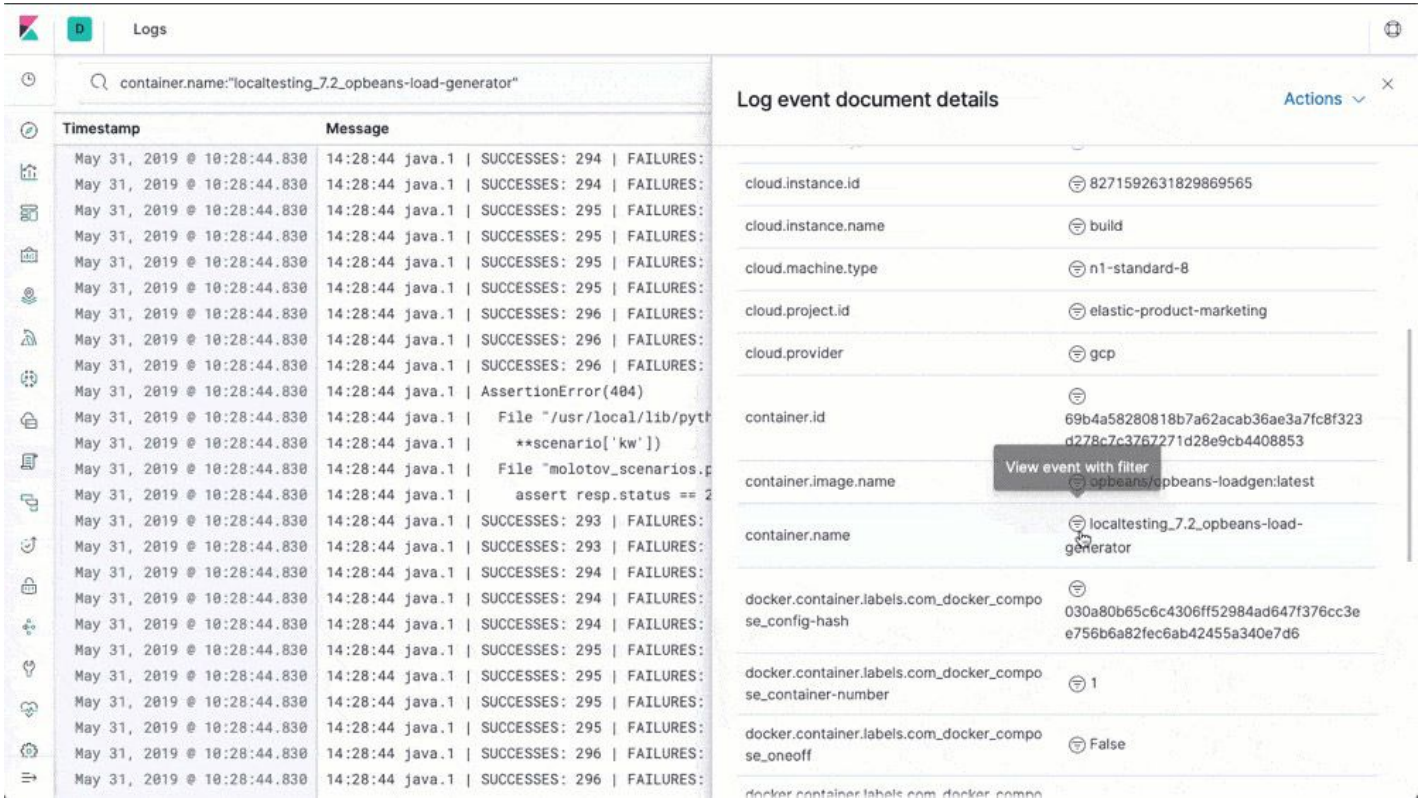
Data visualization: Line chart



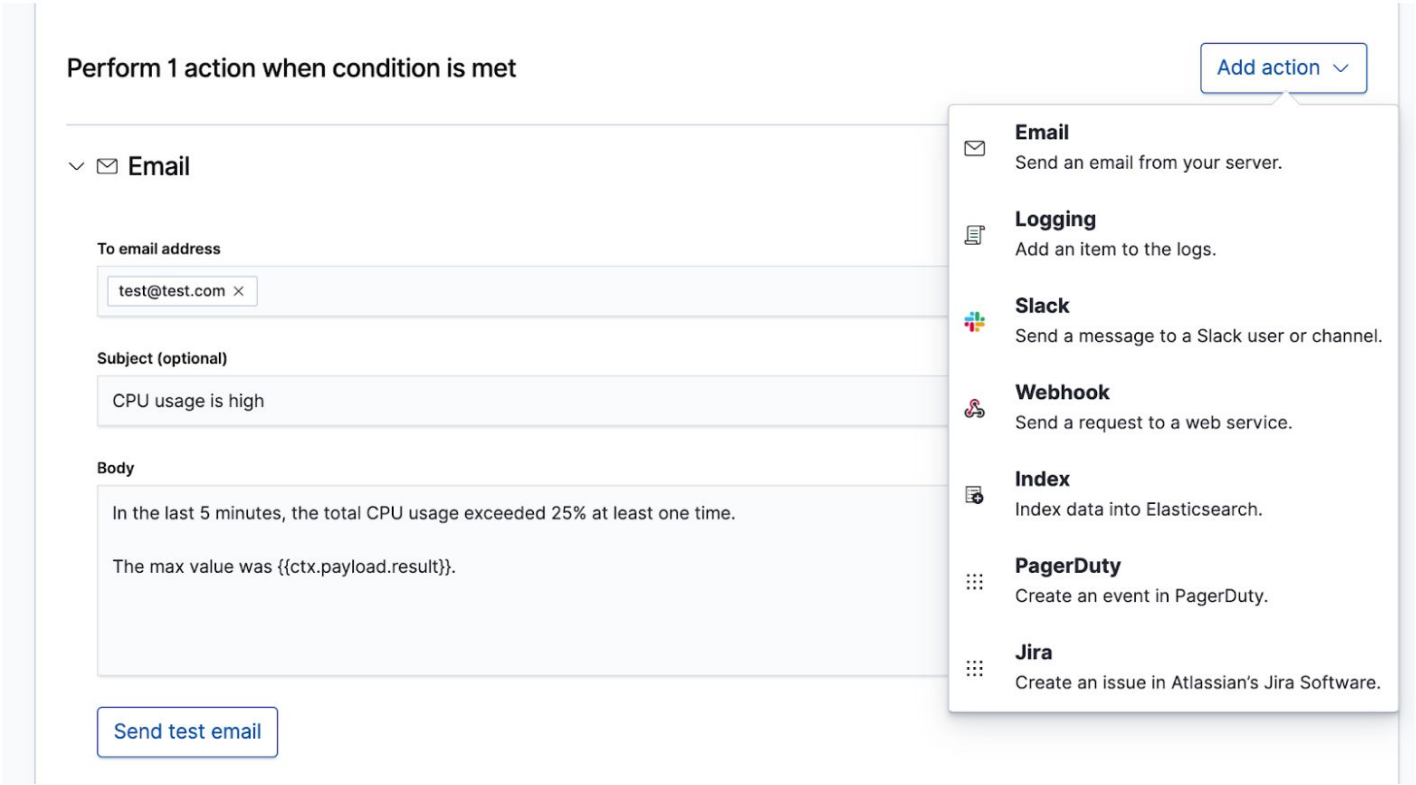
Data visualization: Region map



Monitoring


Logs
container.name:"localtesting_7.2_opbeans-load-generator"
Timestamp
Message
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 294 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 294 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 296 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 296 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 296 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | AssertionError(404)
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | File "/usr/local/lib/pyth
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | **scenario['kw'])
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | File "molotov_scenarios.p
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | assert resp.status == 2
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 293 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 293 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 294 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 294 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 294 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 295 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 296 | FAILURES:
May 31, 2019 @ 10:28:44.830 14:28:44 java.1 | SUCCESSES: 296 | FAILURES:
Log event document details
Actions
cloud.instance.id
8271592631829869565
cloud.instance.name
build
cloud.machine.type
n1-standard-8
cloud.project.id
elastic-product-marketing
cloud.provider
gcp
container.id
69b4a58280818b7a62acab36ae3a7fc8f323d278c7c3767271d28e9cb4408853
View event with filter
container.image.name
opbeans/opbeans-loadgen:latest
container.name
localtesting_7.2_opbeans-load-generator
docker.container.labels.com_docker_compose_config-hash
030a80b65c6c4306ff52984ad647f376cc3ee756b6a82fec6ab42455a340e7d6
docker.container.labels.com_docker_compose_container-number
1
docker.container.labels.com_docker_compose_oneoff
False

Monitoring


Perform 1 action when condition is met
Add action
Email
To email address
test@test.com
Subject (optional)
CPU usage is high
Body
In the last 5 minutes, the total CPU usage exceeded 25% at least one time.
The max value was {{ctx.payload.result}}.
Send test email
Email
Send an email from your server.
Logging
Add an item to the logs.
Slack
Send a message to a Slack user or channel.
Webhook
Send a request to a web service.
Index
Index data into Elasticsearch.
PagerDuty
Create an event in PagerDuty.
Jira
Create an issue in Atlassian's Jira Software.

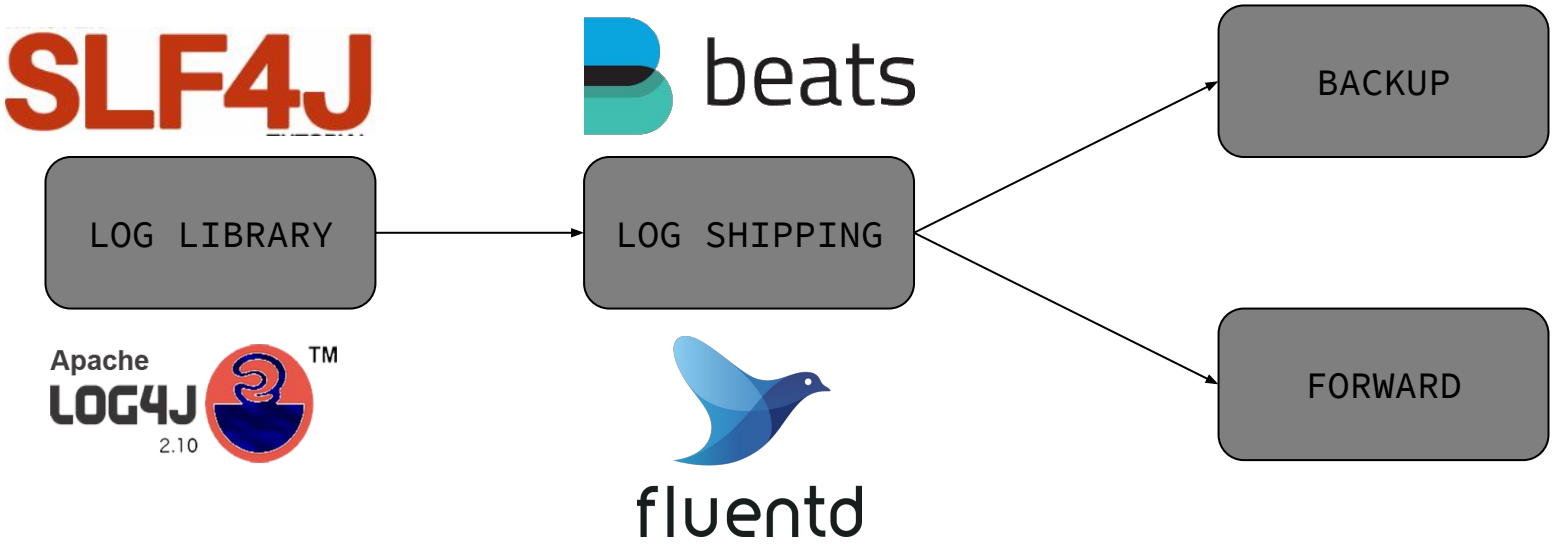
Other features

- Search index
- Canvas
- APM

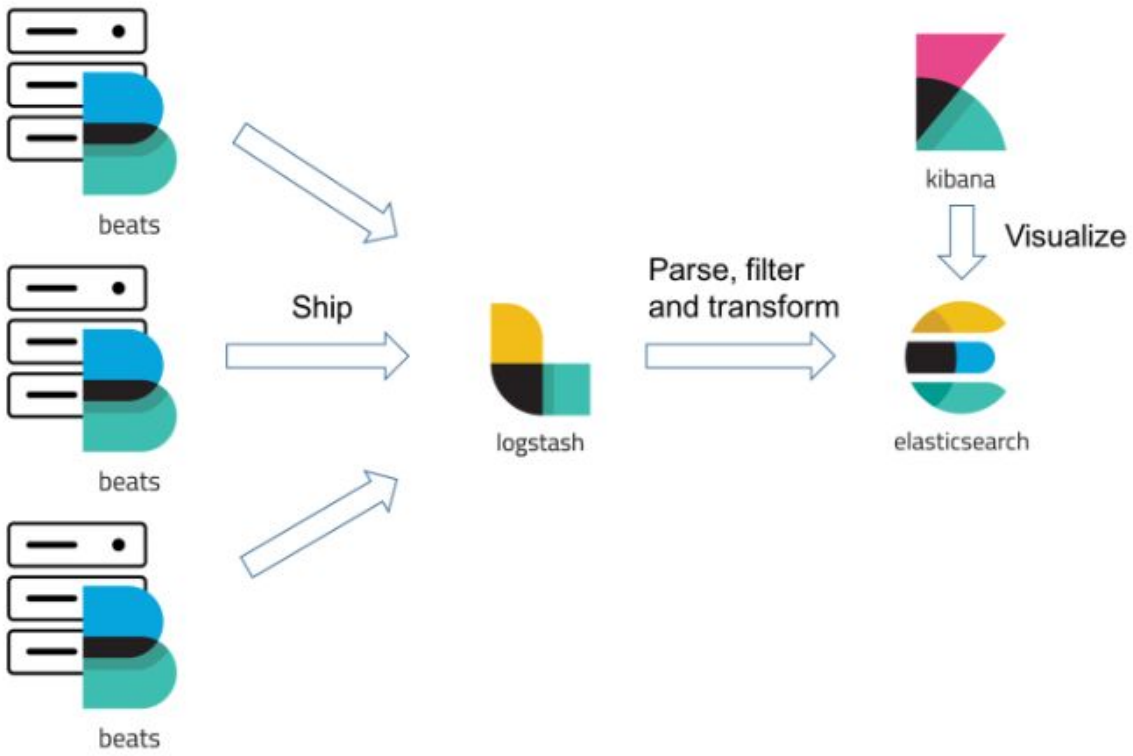
https://drive.google.com/file/d/0B2S_IOa0MiOHWndxWFRiUHNoNW8/view

Log shipper

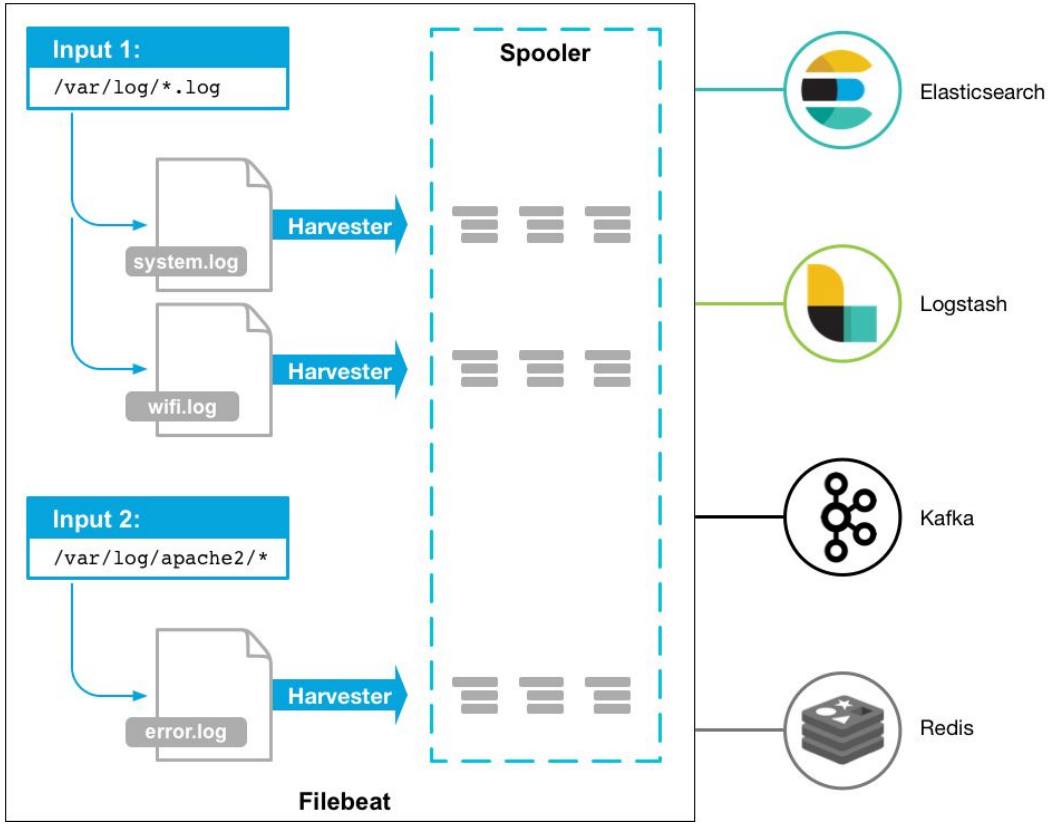
Log shipper (Filebeat, Fluentd, Rsyslog, …)



Filebeat


beats
beats
beats
Ship
logstash
Parse, filter
and transform
kibana
Visualize
elasticsearch

Architecture



Configuration

```
filebeat.inputs:
- type: log
  paths:
  -'/var/lib/docker/containers/*/*.log'
processors:
  - drop_fields:
      fields: ["verb","id"]
output.logstash:
  hosts: ["localhost:5044", "localhost:5045"]
```

Configuration best practice

```
filebeat.inputs:
- type: log
   paths:
    - /var/log/*.log
    - /usr/log/test.log
    - /var/application/logs/*.log
  close_inactive: 10m
  scan_frequency: 20s
  clean_removed: false
  multiline.pattern: '^\['
filebeat.registry:
  path:
    - /var/registry
filebeat.shutdown_timeout: 5s
```

```
output.logstash:
  hosts: ["localhost:5044"]
  loadbalance: true
  worker: 2
  compressive_level: 1
  escape_html: true
  timeoutedit: 20
  bulk_max_size: 2048
  logging.leveledit: debug
```

Thank you

Workshop Team:

@ Tai Tran

@ Hen Tran

@ Vu Nguyen

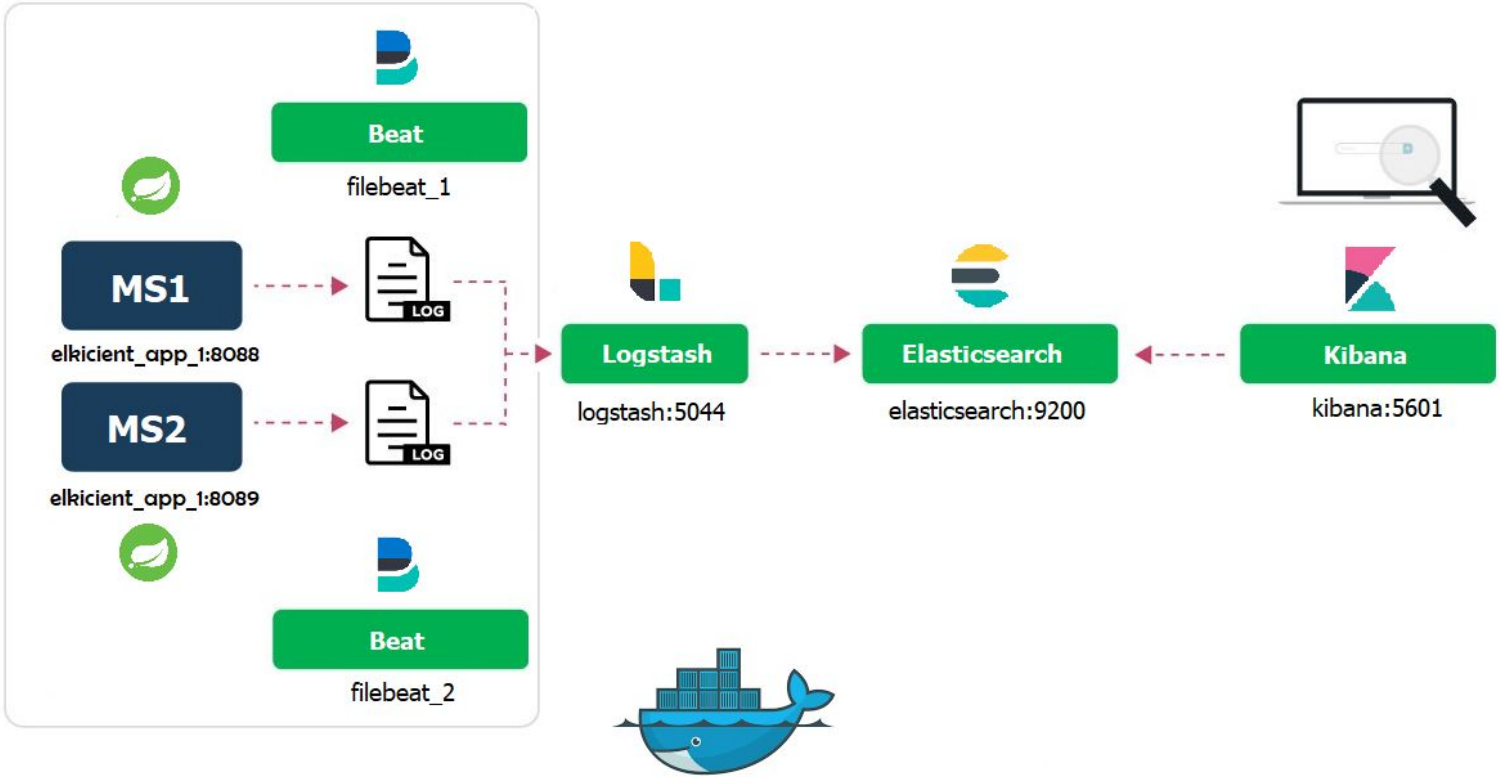
@ Muon Nguyen

@ Phong Pham

@ Dieu Pham

Demo

Infrastructure overview



Hand-on use case

Hand-on use case

Challenges:

- Monitoring the application across multiple environments by Elastic stack
- Centralized logging and visual on dashboard
- Cut off sensitive information
- Notify whenever have errors

Hand-on use case

Time:

- 18:20 - 20:00

Material:

- http://bit.ly/elk_workshop

Resources

Resources

- https://logz.io/learn/complete-guide-elk-stack
- https://www.elastic.co/videos/netflix-using-elasticsearch
- https://www.slideshare.net/Tripwire/my-bro-the-elk?qid=13565215-50a5-4bcb-9d18-11a1e5f0f47e&v=qf1&b=&from_search=25
- https://www.slideshare.net/TinLe1/elk-atlinked-in

Resources

- https://medium.engineering/how-medium-detects-hotspots-in-dynamodb-using-elasticsearch-logstash-and-kibana-aaa3d6632cfd
- https://www.elastic.co/blog/a-full-stack-in-one-command
- https://www.baeldung.com/spring-boot-logging
- https://www.elastic.co/about/history-of-elasticsearch